ARMY RESEARCH LABORATORY

Basin Sculpting a Hybrid Recurrent Feedforward Neural Network

by Michael J. Vrabel

ARL-TR-1522 January 1998

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-1522 **January 1998**

Basin Sculpting a Hybrid Recurrent Feedforward Neural Network

Michael J. Vrabel
Sensors and Electron Devices Directorate

Abstract

The architecture of a recurrent neural-network–based content-addressable memory is detailed along with a companion training algorithm. The memory is designed to store vectors composed of strings of the integers 1 through 9. The performance characteristics of the model—memory capacity and basin size—are presented.

Contents

Figures

1. Introduction

In a seminal paper [1], Hopfield introduced an associative (content-addressable) memory based on a recurrent neural network architecture. Despite the dynamical nature of a recurrent network, convergence to stable memory states for this model is guaranteed. Both the memory capacity of the Hopfield network and the basins of attraction of its memory states have been examined [2,3]. The network was demonstrated to suffer from a limited memory capacity and very irregular basins. Although the original Hopfield model is a binary state device, a model with a multilevel capability was later demonstrated [4], opening the possibility of many other architectural constructs. This report details a more complex neural-network architecture and a companion training algorithm designed to convert the network to a content-addressable memory. The memory is designed to store integer vectors and to have large, error-free basins of attraction.

2. Neural-Network Model

The architecture of the neural network to be described can be characterized as locally feedback (recurrent) and globally feedforward. The neural network is composed of M stages, each containing N nodes. Within each stage, all nodes are fully connected, that is, each node communicates information to all other nodes. A node can exist in one of nine discrete states, defined by the integers 1 through 9. Feedback is incorporated within each stage. The original state of each node n, representing the input data set, is defined as $V_{a,n}^0$. The subsequent states of each node are determined by

$$V_{a,n}^k = \mathcal{F}\left[0.5 + \frac{1}{N}\sum_{m=1}^{N} W_{j,n,m}(V_{a,m}^{k-1})\right], \quad n = 1,\ldots N, \quad k = 1,\ldots 3, \quad (1)$$

where

- a is the designation of the input vector and its children;
- $\mathcal{F}$ indicates that the function is to undergo a real to integer transformation according to the following rule: if $I \leq V_{a,n}^k \leq I + 0.999\ldots$, then $V_{a,n}^k = I$, where I is an integer;
- j is a number, 1 through 9, that takes the value of $V_{a,m}^{k-1}$; thus, the notation $W_{j,n,m}(V_{a,m}^{k-1})$ indicates that the value of $W_{j,n,m}$ (the weight that links nodes n and m) is a function of the $V_{a,m}^{k-1}$ state of node m ($j = V_{a,m}^{k-1}$) (henceforth, I abbreviate the weight term as simply $W_{j,n,m}$);
- $W_{j,n,n} = 0$.

The upper limit on k was chosen because it was found that at $k = 3$, $V_{a,n}^3$ invariably defines a stable point attractor. This is due, in large measure, to the design of the training algorithm.

All neural network stages are cascaded. The output of one stage, that is, the $V_{a,n}^3$ state, is the input to the next stage.

3. Training Algorithm

The neural network is intended to be a content-addressable memory. Stored vectors are to be point attractors. Associated with each attractor is to be a regular, error-free basin of attraction. By *regular* I mean that within some Euclidean distance of each point attractor, all vectors are trapped within its basin.

Training proceeds in several steps. As the evolving family of neural-network weights approaches its final state, increasing levels of sophistication are incorporated into the algorithm. The goal is to determine the relationship between node count and memory storage capacity—the scaling law—and the characteristics of the basins of attraction—primarily, their size. The neural network is trained by being introduced, one at a time, to random vectors chosen to cluster near the stored memories. The neural network weights are then adjusted in response to each probing vector. This procedure continues until the network is fully trained.

To the extent that any neural-network training algorithm can be considered a standard, that standard is backpropagation [5]. In backpropagation, the algorithm defines a measure of error between the actual and desired output of the network and then propagates through the net a small correction to this error (via gradient descent). The recurrent network of the present model has its state updated asynchronously. Because each node is sequentially processed, this process is potentially slow. To speed it up, a major requirement imposed on the training algorithm is that it limits the number of cycles through the net node set before a final, stable attractor state is achieved. We can accomplish this (or, at least, attempt it) by forcing training to proceed not in small increments as with backpropagation, but rather in large increments designed to force the evolving network to an attractor state as rapidly as possible. Unlike backpropagation, training proceeds in a forward direction. The first stage of the network is fully trained, and then the second stage is trained. This procedure continues for all stages. This approach makes it especially convenient to determine the effect of stage count on network performance.

One final note: because the activation function $\mathcal{F}$ of equation (1) is discontinuous, a training algorithm of the backpropagation type—one based

rigorously on gradient descent—is not feasible. The approach taken must be more phenomenological.

3.1 Step One Training

Let b be the designation of one of the vectors to be stored in the neural network. Let i be the designation of a randomly chosen vector. Define the following term:

$$d_{b,i} = \sum_{n=1}^{N} (V_{b,n}^0 - V_{i,n}^3)^2, \tag{2}$$

where $d_{b,i}$ is the square of the distance (Euclidean metric) from memory vector b to test vector i and is the basis for determining the size for the memory basin of attraction. The convention used throughout this report is to designate an attractor state b as $V_{b,n}^0$. The assumption is that, when the network is fully trained, $V_{b,n}^0 = V_{b,n}^1 = V_{b,n}^k$. To simplify all further discussions, I consider only a single component of the above vectors, to be designated n. It is a simple matter to extend all equations to the N terms of the vector.

In the discussion of the training process, several terms need to be defined. First, generate the error value $E_{a,n}^u$:

$$E_{a,n}^u = \sum_{m=1}^{N} W_{j,n,m} - V_{b,n}^0 N. \tag{3}$$

This is the difference between the desired vector n component value of the point attractor, $V_{b,n}^0$, and the component value generated by the randomly chosen vector a scaled by N, the node count. Note that initially feedback effects will be deleted. The initial training algorithm considers only a single pass through equation (1). Thus, the final value of k from equation (1) is to be 1. This is consistent with the desire to force the network into its final attractor state as rapidly as possible.

Several histories are to be created and continually updated. The first is Γ_1:

$$\Gamma_1 = \frac{B_1 \Gamma_1' + \left| E_{a,n}^u \right|}{B_1 + 1}, \tag{4}$$

where Γ_1' is the previously calculated value of Γ_1 and B_1 is a constant. B_1 is used to adjust the time constant of the equation; a typical value is 10. Initially, $\Gamma_1 = 0$. Removed from the beginning of the training algorithm, Γ_1 provides a measure of the present performance of the training algorithm compared to the more recent training cycles.

A parallel set of node weights $\mathcal{W}$ is created. These are updated when $\left|E^u_{a,n}\right| < \Gamma_1$.

$$\mathcal{W}_{j,n,m} = \frac{B_2 \left|E^u_{a,n}\right| \mathcal{W}'_{j,n,m} + W_{j,n,m}}{B_2 \left|E^u_{a,n}\right| + 1} \tag{5}$$

for all m, where $\mathcal{W}'$ is the previously calculated value of $\mathcal{W}$, with $\mathcal{W}$ initialized to 0. B_2 is an adjustable constant typically set to 0.1. When $\left|E^u_{a,n}\right|$ is a small value, B_2 is adjusted so that $B_2 \left|E^u_{a,n}\right|$ is not less than, typically, 5.0. Equation (5) requires such a modification. $\mathcal{W}$ represents an improved weight set when compared with W. These terms are not incorporated into the neural net, but rather are used as part of the training algorithm to periodically adjust the weights of the neural network.

One final item is needed before a discussion of the training algorithm is possible—a running weight average. This is a quantity that is updated with every pass through the algorithm. It reflects, for each node, an adjusted average about the weight values presently being invoked:

$$\mathcal{A}_{i,n,m} = \frac{1}{v'_{n,m}} \sum_{j=1}^{9} v_{j,n,m} W_{j,n,m}, \quad i \neq j, \tag{6}$$

with

$$v_{j,n,m} = \frac{8}{\left|j - V^0_{a,m}\right|} \frac{\theta_{k,n,m}}{\theta'_{n,m}}, \quad j \neq V^0_{a,m}, \tag{7}$$

$$\theta'_{n,m} = \sum_{k=1}^{9} \theta_{k,n,m}, \tag{8}$$

$$v'_{n,m} = \sum_{k=1}^{9} v_{k,n,m}, \tag{9}$$

where $\theta_{k,n,m}$ is an accumulating statistic on the number of times $V^0_{a,m} = k$ for all previous cycles through the training algorithm. $\mathcal{A}_{i,n,m}$ reflects the weight average (connecting nodes n and m) about i, weighted in favor of the points nearest to i and those points that have previously been most often accessed. The logic behind this quantity is as follows: It provides a target for adjusting the weights of the neural network—a target that ensures a relatively smooth weight function as i transitions from 1 through 9. And a smooth weight function helps to minimize the difficulties associated with defining a large basin of attraction.

There is now sufficient introductory material to permit a presentation of the phase one training algorithm. If $E^u_{a,n} > 0$, and $W_{k,n,m} > \mathcal{A}_{k,n,m}$ where $k = V^0_{a,n}$, then if $(W_{k,n,m} - \mathcal{A}_{k,n,m}) < E^u_{a,n}$,

$$E^u_{a,n} = E^u_{a,n} - W_{k,n,m} + \mathcal{A}_{k,n,m} \ , \tag{10}$$

$$W_{k,n,m} = \mathcal{A}_{k,n,m} \ . \tag{11}$$

If $(W_{k,n,m} - \mathcal{A}_{k,n,m}) > E^u_{a,n}$,

$$W_{k,n,m} = W'_{k,n,m} - E^u_{a,n} \tag{12}$$

where the prime indicates the previous (prior to updating) value of $W_{k,n,m}$.

The above process is continued for additional, randomly chosen values of m until either $E^u_{a,n}$ goes to zero (that is, eq (12) is accessed) or all values of m are exhausted. If the above process is exhausted and $\left|E^u_{a,n}\right| d > 0$, then one additional step is required in the phase one training: Assuming $E^u_{a,n} > 0$,

$$W_{k,n,m} = W'_{k,n,m} - \frac{1}{\phi}(W'_{k,n,m} - \mathcal{W}_{k,n,m})E^u_{a,n}, \quad m = 1, \ldots N, \tag{13}$$

where

$$\phi = \sum_{m=1}^{N} (W_{k,n,m} - \mathcal{W}_{k,n,m}) \tag{14}$$

and $W_{k,n,m} > \mathcal{W}_{k,n,m}$.

In equations (10) to (12) above, the algorithm resolves the difference between the output of the neural network and the desired point attractor by forcing the weights via equation (11) or equation (12) to yield the proper output. Where this cannot be fully achieved (based upon the eq (10) to (12) internal criteria), equations (13) and (14) are invoked. This adjusts the weights based on the values of $\mathcal{W}$, the historically best weight fit to the evolving memory. The phase one process is continued until the improvement in network performance, determined by equation (4), asymptotically approaches its final value. At this point, step two training is invoked.

3.2 Steps Two, Three, and Four Training

Step two training involves a more careful analysis of the performance of the network after each step one training cycle and a modification of the standard for calculating $\mathcal{W}$ (eq (5) is disabled). At the end of each step one training cycle, the following additional concepts are invoked. With each training cycle,

the newly calculated weight value is tested (per eq (1)) to a set of C_1 random vectors (vector designation = a). Typically, $C_1 = 20$. The average error is calculated:

$$\mathcal{E}_n^u = \left| \frac{1}{C_1} \sum_{a=1}^{C_1} \left(\mathcal{F} \left[0.5 + \frac{1}{N} \sum_{m=1}^{N} W_{j,n,m} \right] - V_{b,n}^0 \right) \right|. \tag{15}$$

An error history is generated where if $\mathcal{E}_n^u \leq \Gamma_2'$, then

$$\Gamma_2 = \frac{C_2 \Gamma_2' + \mathcal{E}_n^u}{C_2 + 1}, \tag{16}$$

where C_2 is a constant (typical value = 10), and Γ_2' is the previously calculated value of Γ_2. After sufficient training cycles (typically 100) to ensure that equation (16) is no longer dominated by its initialization value, the $\mathcal{W}$ terms can be adjusted: If and only if

$$\mathcal{E}_n^u \leq \Gamma_2', \tag{17}$$

then

$$\mathcal{W}_{j,n,m} = W_{j,n,m}. \tag{18}$$

If after typically 200 training cycles, equation (17) is not satisfied, then

$$W_{j,n,m} = \mathcal{W}_{j,n,m} \tag{19}$$

for all j, n, and m, and training continues. As this training phase approaches a limiting value (based on the eq (16) results), the third step of training is entered. The third step is identical to the second step, except that the error term of equation (15) is based on a single level of feedback. That is, for equation (1) the final value for k is raised from 1 to 2. The fourth step of training is identical to the third, except that the final value for k is 3.

4. Neural Network Performance

Figures 1 to 3 summarize the performance of the neural network and its training algorithm. On these figures, *error rate* refers to the failure of a randomly chosen vector to converge to the closest memory (or point attractor). A basin edge is defined by that set of points equidistant between some memory state and its nearest neighbor. Each curve can then be interpreted as an average shape of the basin of attraction about each intentional point attractor.

Each memory set and test vector is chosen randomly. All memory sets were selected from a large base of vectors. The members of each set were chosen to be as distant as practical (based on the distance definition of eq (2)) from all other members, based on a random selection process. These figures demonstrate what was observed throughout the algorithm test phase: model performance is substantially independent of the vector set used. The curves of figures 1 to 3 (the results of a single, randomly chosen family of attractor states) are a reasonable representation of the performance of the neural network and its training algorithm.

The results in figures 1 to 3 can be summarized succinctly. Figure 1 demonstrates the expected result—as the memory count increases, the basin sizes shrink. Figures 2 and 3 demonstrate that little is to be gained from using more than three stages. Figures 2 and 3 also demonstrate that memory capacity scales poorly (if at all) with an increased vector length—at least, in the range of N up to 15. The curves also demonstrate that, depending on memory count, each memory can have a sizable error-free basin of attraction.

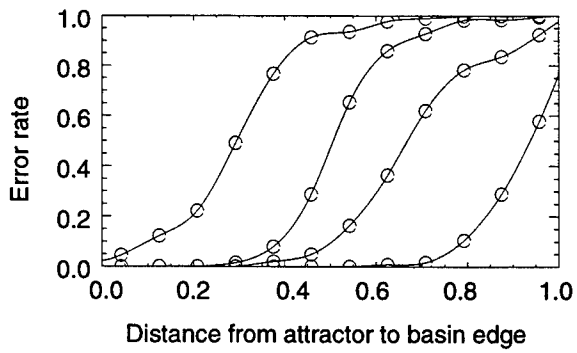

Figure 1. Basin performance as function of memory count. Memory count = 2 to 5 and vector length = 12.

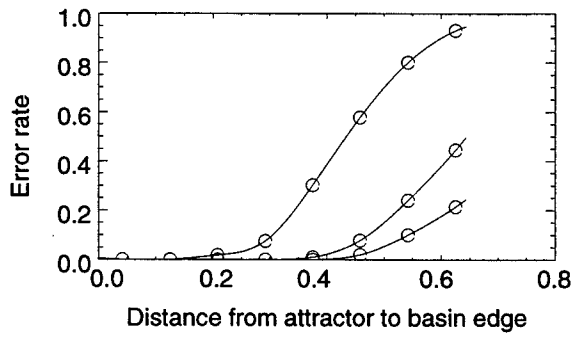

Figure 2. Basin performance as function of stage count. Memory count = 3, vector length = 8, and stage count = 1 to 3.

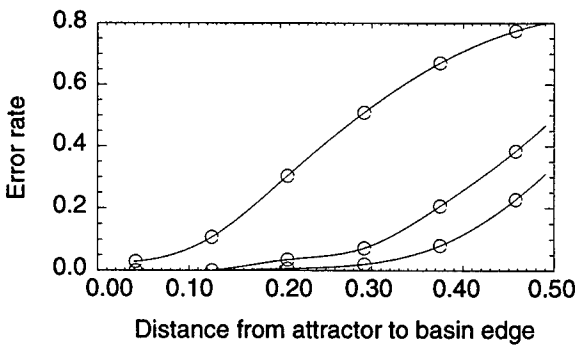

Figure 3. Basin performance as function of stage count. Memory count = 4, vector length = 15, and stage count = 1 to 3.

5. Conclusion

I have demonstrated the feasibility of creating a multistate (as opposed to binary) multistage neural-network–based content-addressable memory and training algorithm. The model can generate large basins of attraction, albeit for a very limited number of attractor states.

References

1. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," Proc. Nat. Acad. Sci **79** (April 1982), 2554–2558.

2. R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The capacity of the Hopfield associative memory," IEEE Trans. Inf. Theory **IT-33** (July 1987), 461–482.

3. J. D. Keeler, "Basins of attraction of neural network models," AIP Conf. Proc. 151, *Neural Networks for Computing*, Snowbird, UT (1986), pp 259–264.

4. M. Fleisher, "The Hopfield model with multi-level neurons," *AIP Neural Information Processing Systems*, Denver, CO (1987), pp 278–289.

5. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature London **323** (1986), 533–536.

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Ofc of the Dir Rsrch and Engrg
Attn R Menz
Pentagon Rm 3E1089
Washington DC 20301-3080

Ofc of the Secy of Defns
Attn ODDRE (R&AT) G Singley
Attn ODDRE (R&AT) S Gontarek
The Pentagon
Washington DC 20301-3080

OSD
Attn OUSD(A&T)/ODDDR&E(R) J Lupo
Washington DC 20301-7100

CECOM
Attn PM GPS COL S Young
FT Monmouth NJ 07703

CECOM RDEC Elect System Div Dir
Attn J Niemela
FT Monmouth NJ 07703

CECOM
Sp & Terrestrial Commctn Div
Attn AMSEL-RD-ST-MC-M H Soicher
FT Monmouth NJ 07703-5203

Dir of Assessment and Eval
Attn SARD-ZD H K Fallin Jr
103 Army Pentagon Rm 2E673
Washington DC 20301-0163

Hdqtrs Dept of the Army
Attn DAMO-FDT D Schmidt
400 Army Pentagon Rm 3C514
Washington DC 20301-0460

MICOM RDEC
Attn AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

US Army Avn Rsrch, Dev, & Engrg Ctr
Attn T L House
4300 Goodfellow Blvd
St Louis MO 63120-1798

US Army CECOM
Night Vision & Elec Sensors Dir
Attn AMSEL-RD-NV-VISPD C Hoover
10221 Burbeck Rd, Ste 430
FT Belvoir VA 22060-5806

US Army CECOM Rsrch, Dev, & Engrg
Attn R F Giordano
FT Monmouth NJ 07703-5201

US Army Edgewood Rsrch, Dev, & Engrg Ctr
Attn SCBRD-TD J Vervier
Aberdeen Proving Ground MD 21010-5423

US Army Info Sys Engrg Cmnd
Attn ASQB-OTD F Jenia
FT Huachuca AZ 85613-5300

US Army Materiel Sys Analysis Agency
Attn AMXSY-D J McCarthy
Aberdeen Proving Ground MD 21005-5071

US Army Matl Cmnd
Dpty CG for RDE Hdqtrs
Attn AMCRD BG Beauchamp
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Matl Cmnd
Prin Dpty for Acquisition Hdqrts
Attn AMCDCG-A D Adams
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Matl Cmnd
Prin Dpty for Techlgy Hdqrts
Attn AMCDCG-T M Fisette
5001 Eisenhower Ave
Alexandria VA 22333-0001

US Army Natick Rsrch, Dev, & Engrg Ctr
Acting Techl Dir
Attn SSCNC-T P Brandler
Natick MA 01760-5002

Distribution (cont'd)

US Army Rsrch Ofc
Attn G Iafrate
4300 S Miami Blvd
Research Triangle Park NC 27709

US Army Simulation, Train, & Instrmntn Cmnd
Attn J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army Tank-Automtv & Armaments Cmnd
Attn AMSTA-AR-TD C Spinelli
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Tank-Automtv Cmnd Rsrch, Dev, & Engrg Ctr
Attn AMSTA-TA J Chapin
Warren MI 48397-5000

US Army Test & Eval Cmnd
Attn R G Pollard III
Aberdeen Proving Ground MD 21005-5055

US Army Train & Doctrine Cmnd Battle Lab Integration & Techl Dirctrt
Attn ATCD-B J A Klevecz
FT Monroe VA 23651-5850

US Military Academy
Dept of Mathematical Sci
Attn MAJ D Engen
West Point NY 10996

USAASA
Attn MOAS-AI W Parron
9325 Gunston Rd Ste N319
FT Belvoir VA 22060-5582

Nav Surface Warfare Ctr
Attn Code B07 J Pennella
17320 Dahlgren Rd Bldg 1470 Rm 1101
Dahlgren VA 22448-5100

GPS Joint Prog Ofc Dir
Attn COL J Clay
2435 Vela Way Ste 1613
Los Angeles AFB CA 90245-5500

Special Assist to the Wing Cmndr
Attn 50SW/CCX Capt P H Bernstein
300 O'Malley Ave Ste 20
Falcon AFB CO 80912-3020

DARPA
Attn B Kaspar
Attn L Stotts
3701 N Fairfax Dr
Arlington VA 22203-1714

ARL Electromag Group
Attn Campus Mail Code F0250 A Tucker
University of Texas
Austin TX 78712

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Prsnnl
Attn J Hiller
The Pentagon Rm 2C733
Washington DC 20301-0300

ERIM
Attn J Ackenhusen
1975 Green Rd
Ann Arbor MI 48105

US Army Rsrch Lab
Attn AMSRL-CI-LL Techl Lib (3 copies)
Attn AMSRL-CS-AL-TA Mail & Records Mgmt
Attn AMSRL-CS-AL-TP Techl Pub (3 copies)
Attn AMSRL-SE J M Miller
Attn AMSRL-SE J Pellegrino
Attn AMSRL-SE-SA J Eicke
Attn AMSRL-SE-SE D Nguyen
Attn AMSRL-SE-SE M Vrabel (10 copies)
Attn AMSRL-SE-SE T Kipp
Attn AMSRL-SE-SR D Rodkey
Adelphi MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1998	Final, from Oct 1996 to June 1997

4. TITLE AND SUBTITLE
Basin Sculpting a Hybrid Recurrent Feedforward Neural Network

5. FUNDING NUMBERS
PE: 61102A
DA PR: A305

6. AUTHOR(S)
Michael J. Vrabel

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL-SE-SE (vrabel@arl.mil)
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1522

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
AMS code: 611102.305
ARL PR: 7NE0M1

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The architecture of a recurrent neural-network–based content-addressable memory is detailed along with companion training algorithm. The memory is designed to store vectors composed of strings of the integers 1 through 9. The performance characteristics of the model—memory capacity and basin size—are presented.

14. SUBJECT TERMS
Associative memory, content-addressable memory, recurrent neural network

15. NUMBER OF PAGES
21

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102